# CS-308-2014 Final Report

**Target Destroyer**
**TH-11**

**Pravesh Agrawal, 100050060**
**Abhie Shah, 100050002**
**K Daya Shankar Reddy, 100050070**
**Uday Shah, 100050064**
**Kartik Chaudhary, 100050019**

## Table of Contents

# 1. Introduction

We are building a robot, which given a pre-identifiable target, fires bullets at it. Note that human interaction is limited to refilling bullets and recharging batteries. Motion, identification of target, aim and shooting are all automated.

# 2. Problem Statement

Given a series of targets in a line, identify those targets and shoot them. The entire operation should be automated. A semi-automatic gun (reloads automatically), cameras and motors are provided. This project has various applications, some of which include targeting birds/pests in fields and drone strikes for the army.

# 3. Requirements

## 3.1 Functional Requirements

The exact requirements are in bold. Surrounding text is basic explanation. Our bot meets the following:

1. Motion: **The bot moves randomly along pre-defined paths.** Currently, we use a depth-first search algorithm, but as targets are expected to be generated randomly in the real world, this has no benefit over any other implementation.

2. Identification: We use red balls as targets (code is easily modifiable for different usage). **The camera on the robot continuously analyses its input, and detects these.** The code used for this is elaborated on later in this document.

3. Aim: We are using a typical Chinese air-gun available in the market. Two motors are needed. The first moves the gun in the horizontal direction (forwards and backwards) and the second in a vertical direction (in terms of angle). **The bot calculates the angle required based on the analysis using the motors and settles it in a position ready for shooting.** Note that power is inconsequential here since we can neglect gravity and air resistance with respect to the bullet velocity. The gun, anyway, has no power option.

4. Shooting: **The gun fires and shoots the target.** We use a rubber band for pulling the trigger. Alternative mechanisms may be considered.

For a successful shot, all these four individual requirements must be fulfilled.

### 3.2 Non-Functional Requirements

Nothing per se, talked about when required under other sections

### 3.3 Hardware Requirements

The following are in addition to the kit provided (red font means addition from SRS:

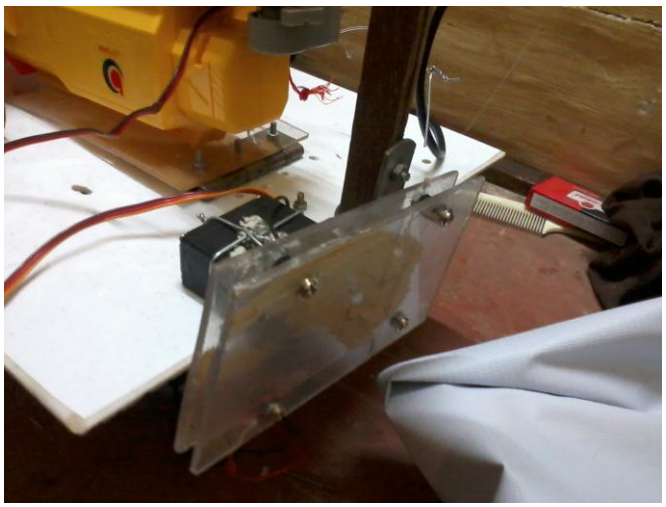
1 Chinese air gun with plastic pellets (bought from the market)(semi-automatic for ease in reloading)
1 Camera (provided by the lab team)
1 Servo Motor for angle adjustment (provided by the lab team)
1 DC Motor for horizontal motion (provided by the lab team)
Rubber bands, wood and basic building material for the gun (bought from the market)
2 pulleys, Steel wire

### 3.4 Software Requirements

Image processing to be done through Python. Other motion and interfacing is basic code already covered.

## 4. System Design

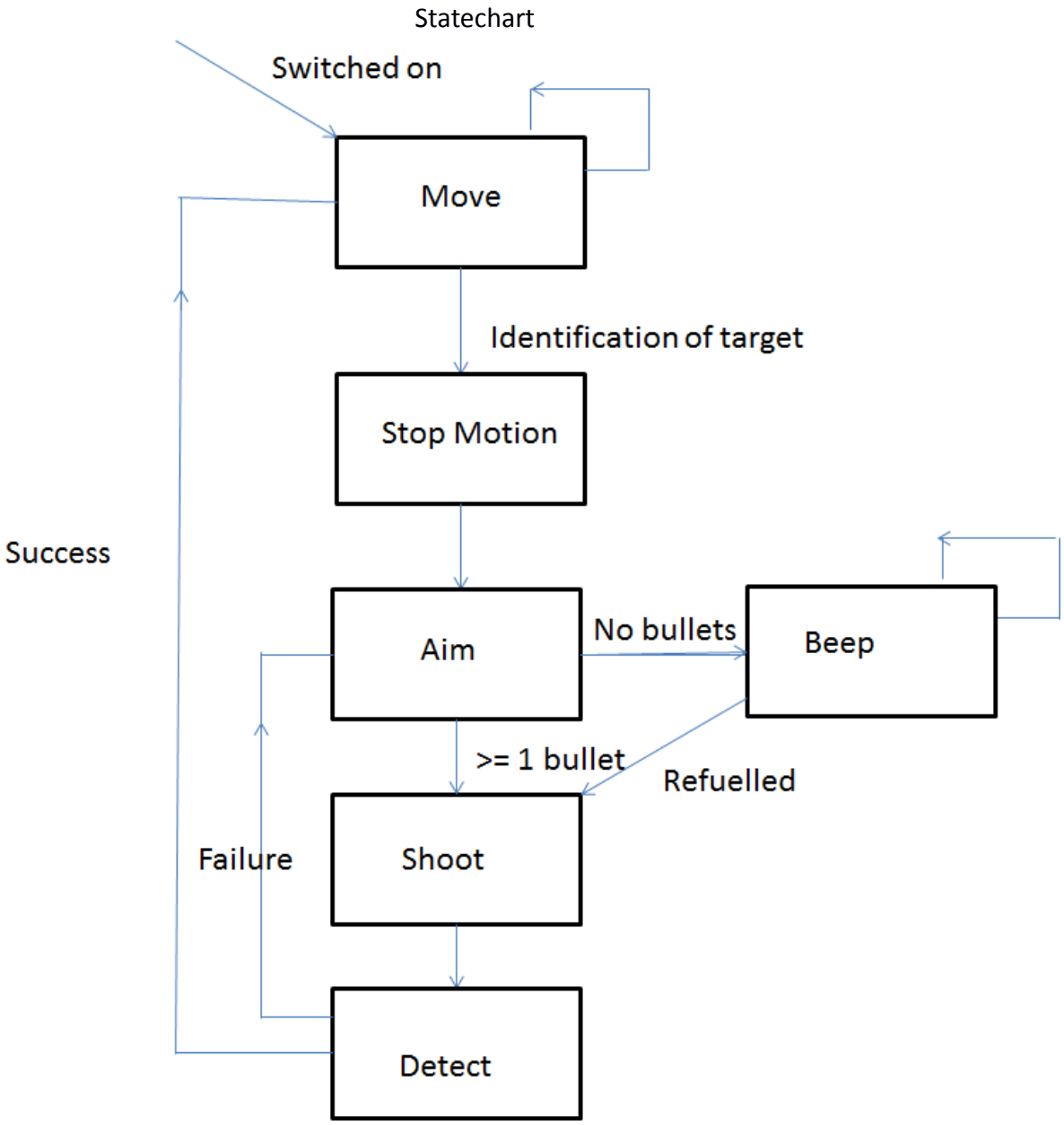
The picture above shows our contraption. Note the servo motor on top of the gun to pull the trigger.



This servo motor is responsible for lifting the gun. It is placed so that the thread coming out of it is parallel to the pulley on the wooden block.

The pulley mechanism shown here.

Camera attached to the barrel.




## 5. Working of the System and Test results

We had proposed a testing strategy, we are filling in our results.

Assume the existence of a feasible number T for the number of successful shots in 3 minutes, given the presence of a high density of targets. If our bot crosses T, it is successful. We can settle upon a suitable value based on our initial testing.

This method is extremely harsh, and we propose an alternative to break it up into the four steps. Each step assumes that the previous are successful. (If they don't work, we will manually achieve those by inputting appropriate data and/or physically adjusting the robot.) I, A and S are numbers to be finalized upon after testing. We expect I and A to be close to 0.9, S to be close to 0.75.

After testing we find that, putting aside extra manual time, once it identifies the image, the bot takes a maximum of 8 seconds to shoot. I is 1 if no external interference is present. A has a high variance depending upon several factors, the most important of which is battery power. At a distance of 5 feet, we find that all shots fall within 7 cm of the target. (Note that this is AFTER calibration). Our motor could not pull the trigger. It tries to, but a little push is needed at the end. We, therefore, do not quantify S.

| Requirement | Strategy | Results |
|---|---|---|
| **Motion** | Check if the robot is able to detect obstacles, and move accordingly. | Satisfactory |
| **Identification** | Find the value of number of targets identified/ number of targets encountered. Value should be no lesser than I. | Target always identified |
| **Aim** | Once the aim is finalized, manually trigger the gun. Find the value of targets hit/ targets shot at. Value should be no lesser than A. | After calibration, bullet falls within an acceptable margin. Rigorous calibration required for better results. |
| **Shooting** | Put the gun in a position where manual shooting will result in a successful shot. Now, let the robot shoot. Find the value of targets hit/targets shot at. Value should be no lesser than S. | Not Quantified. |

Working of System covered in the previous and next sections.

# 6. Discussion of System

**We start off with the mistakes/changes we made.**

1. **Size of Gun and its consequences.**

We, rather foolishly, did not confirm the dimensions of our gun before placing the order. It turned out to be over twice the size we expected. This led to a complete overhaul in our design. Firstly, we

had to abandon the two-camera plan as the barrel of the gun interfered with their view, no matter where the cameras were placed. We decided to place a single camera near the barrel to solve the issue.

2. **Placement of Motors**

Our original plan was to have both motors, for lifting and for triggering, on the acrylic board. This led to unstability, as the orientation of the gun was changed on the horizontal axis too, which consequently led to inaccuracy. To solve this, we placed the motor for lifting in line with the axis of the gun. The motor for triggering had to be bolted on to the gun itself. (See image).

3. **Orientation of gun**

We had initially planned to place the gun in the forward direction. This led to the problem that images only in line with the axis could be shot. Else, we had to go to the right (or left) and turn again, process the new image, and then repeat till it is in line. This is time-consuming when we have a pre-defined grid like in a greenhouse. We decided to place the gun to the left, and stop when the target is at the center. Again, even if there are small disturbances (bot is not traveling in a line), as the gun is rigid with respect to the bot, we can still shoot targets effectively. The problem here is that we are effectively blind in the forward and right direction. To solve this, we propose that teams in the future use a servo motor to rotate the board itself, allowing for vigilance in all directions (mentioned in future work).

**Components that worked according to plan:**

Our basic idea of having a string and board contraption to lift the gun was successful. Despite the weight and size of the gun, the servo motor was strong enough to pull the gun up. On the programming aspect, we were able to accomplish what we set out to do and the gun was calibrated after some test rounds. We had certain doubts regarding whether we needed a bot which could process images itself, without the need for the computer, as we thought that image transmission to the computer and the consequent commands may take time, but we found the response time satisfactory.

**Things not in SRS which were added:**

A major problem we faced was the lateral movement of the gun while it was being lifted. For this, on top of the steps mentioned above, we decided to add a thin layer of acrylic sheet beneath the gun, and made the necessary dents in it so that the gun could fit in (rather than just be stuck with glue). This created a lock-type mechanism, which reinforced with steel wires, made the gun stable.

# 7. Future Work

We have set up a frame-work upon which more complex and interesting projects can be built. The code is easy to re-use and has the option of implementation of functions. Our bot currently shoots bullets at stationary blue objects. The next step would be to target objects moving in a straight

line. After that, we can have image detection for random paths. Also, the placement of another servo motor beneath the board to allow it a 180 degree view may be considered. (Not too much effort on the mechanical side, but coding may be a challenge.)

An important limiting factor with the current project is the time taken to shoot after image locking. Our bot, in the worst case, takes about 8 seconds to shoot. This needs to be improved upon in two ways. Firstly, at the risk of a marginal error in the angle, we can increase the speed of lifting the gun. Secondly, the trigger mechanism needs to be looked at. The motor used currently takes a lot of time, and this mechanism cannot be used while considering fast-moving targets.

Possible applications of our project include:

1. Robots to shoot pests (specifically slightly large targets like rats) in farms
2. Building upon the already prevalent drones, we can have extremely small and mobile guns to be sent for scouting and securing regions.

# 8. Conclusions

We succeeded in accomplishing our original idea, but feel that a lot of work can be done on top of what has been achieved. Our final robot looked very, very different from what we had in mind, and we'd recommend that teams in the future be sure about all aspects of their project, and test these components individually before putting them together.

# 9. References

Python OpenCV Library- squares.py for recognition of squares in an image.